CSC 411

Computer Organization (Spring 2023)
Lecture 1: Logistics

Prof. Marco Alvarez, University of Rhode Island

Welcome !

- Lectures
  - TTh 5:30-6:45p @ Eng 045
- Lab
  - W 5-5:50p @ Zoom
- Office Hours
  - W 4-5p @ Zoom
- Team
  - Marco Alvarez, Instructor
  - TBA
- Course Website
  - https://homepage.cs.uri.edu/~malvarez/teaching/csc-411/

What is this course about?

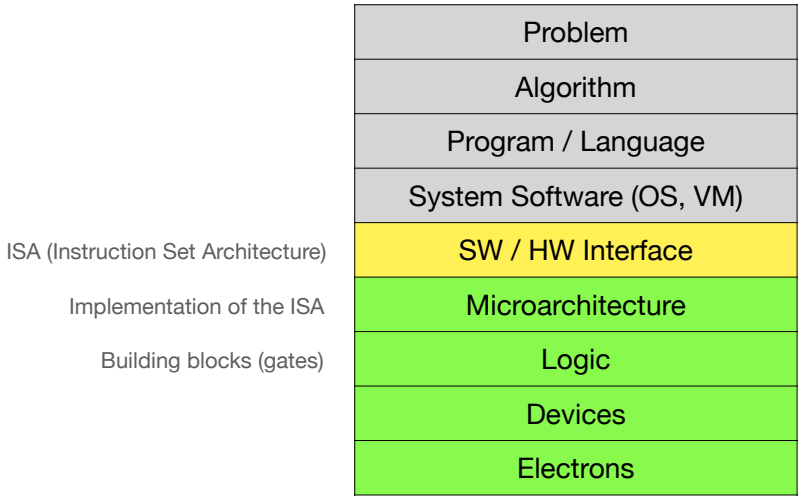
CSC 411: Computer Organization
LEC: (4 crs.) Logical structure of computer systems viewed as a hierarchy of levels. Assembly language programming, assemblers, linkers, loaders. Computer architecture including digital logic, processor organization, instruction sets, addressing techniques, virtual memory, microprogramming. (Lec. 3, Project 3) Pre: CSC 212 and student must be admitted to a degree-granting college.

How computers work?

Computers?

- Why do we have computers?
  - to solve problems, gain insight
- How do computers solve problems?
  - moving electrons around

A computing system



What is computer architecture?

- Science of designing and implementing computing systems
  - HW/SW interface and below
  - expanding to upper layers
- Design goals
  - highest performance
  - optimizing for energy efficiency
  - best performance/cost ratio
- Think about design goals for a supercomputer vs an smartphone
  - fundamental principles are similar

Why study computer architecture?

- Understanding current/future capabilities of computing systems
  - why computers work the way they do
- Developing better software
  - best system programmers understand all abstraction levels and the underlying hardware
- Understanding computer performance
  - writing well tuned software requires knowing what's under the hood
- Setting up the fundamentals for further work on hardware design

To illustrate the potential gains from performance engineering, consider multiplying two 4096-by-4096 matrices. Here is the four-line kernel of the Python 2 code for matrix-multiplication:

```
for i in xrange(4096):
    for j in xrange(4096):
        for k in xrange(4096):
            C[i][j] += A[i][k] * B[k][j]
```

Version	Implementation	Running time (s)	GFLOPS	Absolute speedup	Relative speedup	Fraction of peak (%)
1	Python	25,552.48	0.005	1	–	0.00
2	Java	2,372.68	0.058	11	10.8	0.01
3	C	542.67	0.253	47	4.4	0.03
4	Parallel loops	69.80	1.969	366	7.8	0.24
5	Parallel divide and conquer	3.80	36.180	6,727	18.4	4.33
6	plus vectorization	1.10	124.914	23,224	3.5	14.96
7	plus AVX intrinsics	0.41	337.812	62,806	2.7	40.45

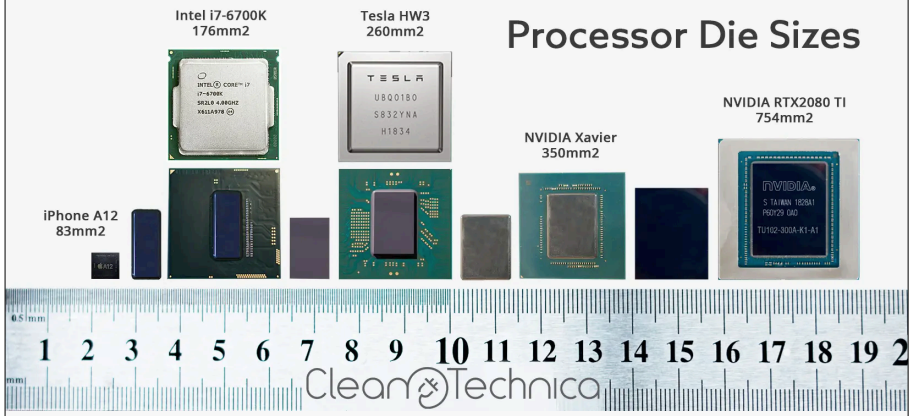
From: "There's plenty of room at the Top: What will drive computer performance after Moore's law? "

Modern computer architecture

- Achieving higher performance and efficiency
  - **co-design** across the hierarchy (bring algorithms to devices)
  - **specialize** as much as possible
- Looking forward …
  - same basic building blocks and design principles
- Exciting times in computer architecture
  - novel architectures
  - renovated and powerful computing landscape

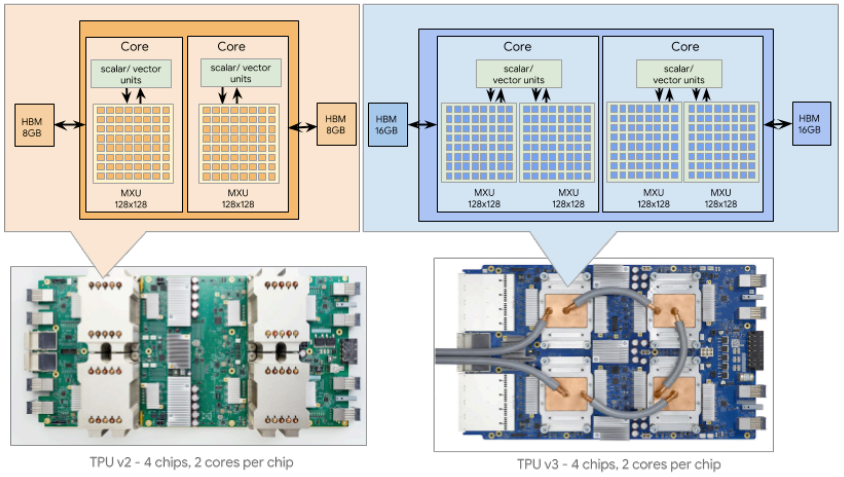
Example platforms

Getting a sense of size



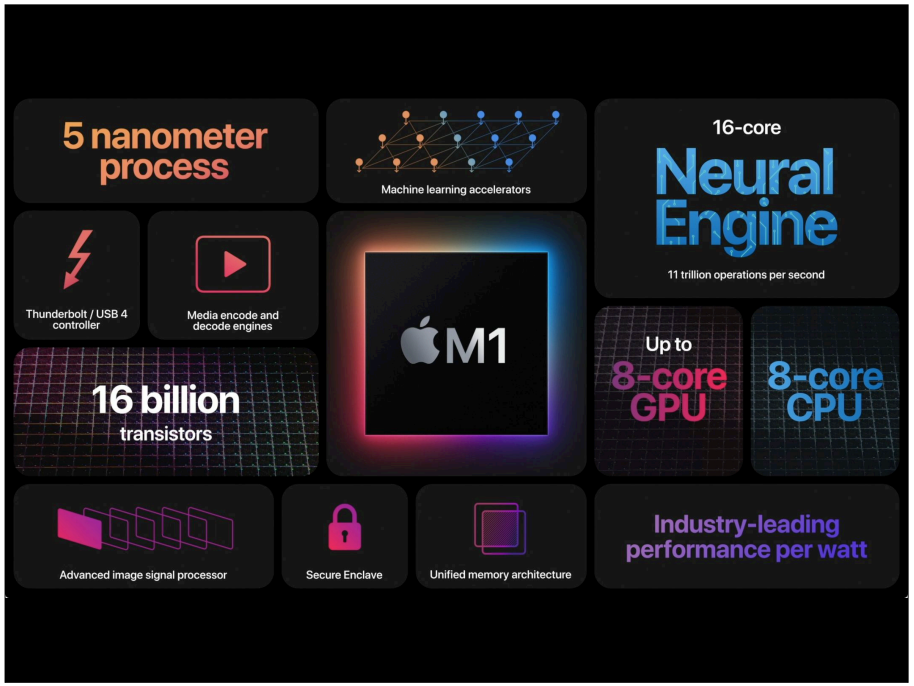
https://cleantechnica.com/2019/06/15/teslas-new-hw3-self-driving-computer-its-a-beast-cleantechnica-deep-dive/

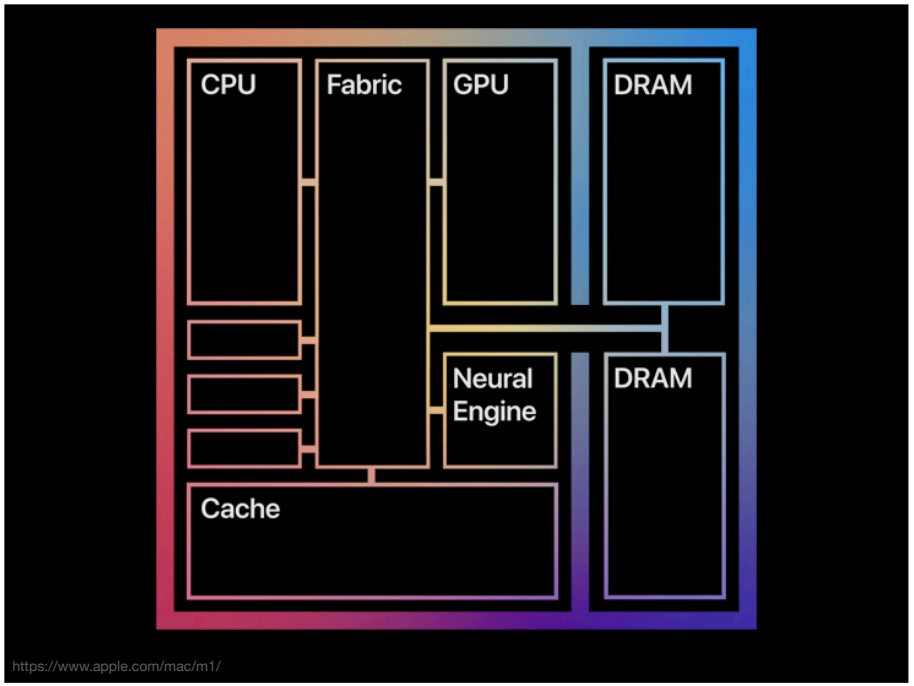
TPU (Tensor Processing Unit)



TPU v2 - 4 chips, 2 cores per chip

TPU v3 - 4 chips, 2 cores per chip



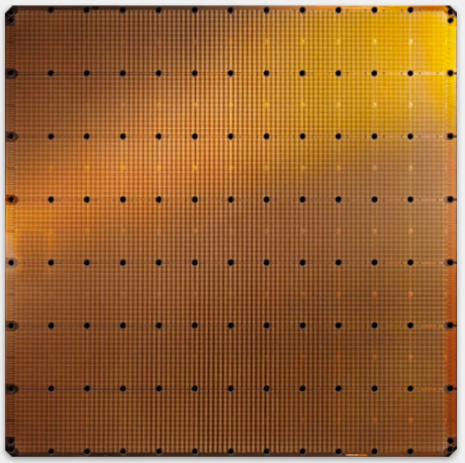


https://www.apple.com/mac/m1/

Fugaku

Fugaku remains the No. 1 system. It has 7,630,848 cores which allowed it to achieve an HPL benchmark score of 442 Pflop/s. This puts it 3x ahead of the No. 2 system in the list. Fujitsu RIKEN Center for Computational Science, Japan.

https://www.top500.org/lists/top500/2021/11/



Cerebras WSE-2
46,225mm² Silicon
2.6 Trillion transistors

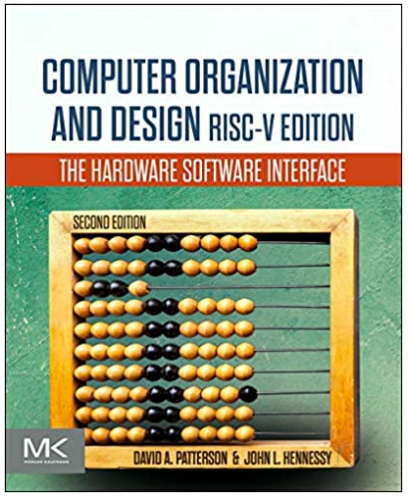
Largest GPU
826mm² Silicon
54.2 Billion transistors

https://www.cerebras.net/product-chip/

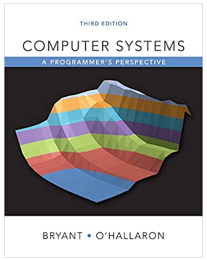
Cerebras's Wafer Scale ML Engine

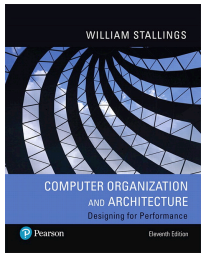
Course organization

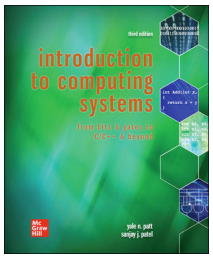
Recommended textbooks



Required







Grading

- Assignments (25%)
  - programming (primarily C and RISC-V)
  - problem sets (mostly from the textbook)
- Technical Presentation (25%)
  - teams of 2
- Exams
  - 1 midterm (25%)
  - 1 final (25%)

gradescope

Assignments

- Discussions and collaboration are allowed
  - you must write your own code and solutions
- Late submissions **NOT** accepted
  - ample time given for assignments (~8 days)
  - start and submit early, leaving plenty of time for updates
- Plagiarism?
  - just don't do it
  - reports are sent to the chair with copies to your dean, the student's dean, and the office of student life

How to succeed?

- Attend all lectures/labs
  - lectures run synchronously and are not being recorded
  - attendance usually correlates with higher grades
- Participate and think critically
  - turn on your cameras during lectures and feel free to ask questions
  - use the online forum (EdStem)
  - use office hours regularly
- Start working on assignments early
  - avoid copying/pasting or google'ing answers